

Depth Estimation using Modified Cost Function for Occlusion Handling

Krzysztof Wegner and Olgierd Stankiewicz

Abstract—The paper presents a novel approach to occlusion handling problem in depth estimation using three views. A solution based on modification of similarity cost function is proposed. During the depth estimation via optimization algorithms like Graph Cut similarity metric is constantly updated so that only non-occluded fragments in side views are considered. At each iteration of the algorithm non-occluded fragments are detected based on side view virtual depth maps synthesized from the best currently estimated depth map of the center view. Then similarity metric is updated for correspondence search only in non-occluded regions of the side views. The experimental results, conducted on well-known 3D video test sequences, have proved that the depth maps estimated with the proposed approach provide about 1.25 dB virtual view quality improvement in comparison to the virtual view synthesized based on depth maps generated by the state-of-the-art MPEG Depth Estimation Reference Software.

Keywords—depth estimation, disparity estimation, occlusion handling, MVD, graph cuts, DERS, Free viewpoint television.

I. INTRODUCTION

3D video systems have recently gained a lot of attention. Many new 3D video systems have been developed. Among them super multiview television and free viewpoint television can be examples of such novel 3D systems. In the free viewpoint television a user is able to freely choose a position of a virtual camera. The requested view of a scene is generated from dynamic 3D representation of the scene.

The most commonly used 3D representation is a MultiVideo and Depth (MVD) [6] composed of multiple videos acquired by the set of cameras and accompanied depth maps for each of the views. Based on transmitted videos and depth data any view can be easily generated by employing depth-image-base rendering (DIBR) [7].

Recently 3D extension of such standards as AVC [32], [33] and HEVC [31] that allows for efficient transmission of dynamic 3D scene representation in MVD format has been finalized.

Depth information in such systems can be acquired either directly by depth cameras [8], or indirectly by algorithmic depth estimation from recorded videos [9]. Commonly depth information is obtained by the conversion from disparity information [10]. Although in computer vision, disparity d is often treated as synonymous with depth (distance z), essentially those terms are the inverse of each other.

$$z \sim \frac{1}{d} \quad (1)$$

Krzysztof Wegner and Olgierd Stankiewicz are with the Chair of Multimedia Telecommunications and Microelectronics, Poznan University of Technology, Poznan, Poland (e-mail: kwegner@multimedia.edu.pl)

Disparity is a displacement vector between corresponding fragments (pixels, blocks) of two images of the same scene taken from different viewpoints. Those two corresponding fragments represent the same fragment of an observed scene but seen from two different viewpoints.

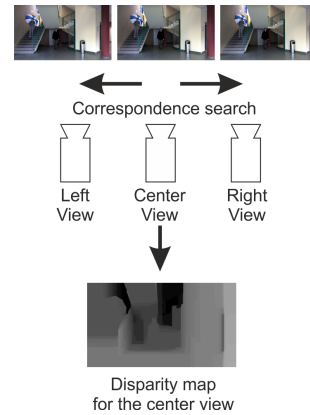


Fig. 1. Three-view disparity estimation.

Stereo correspondence search is an active research topic in computer vision, and one of the basic method of obtaining disparity information. There are many stereo disparity estimation methods known. Comprehensive study of stereo disparity estimation methods can be found in [34], and on the Middlebury webpage [30] containing up-to-date benchmark of stereo disparity estimation methods. In the scope of development of multiview systems stereo correspondence search was extended to multiview correspondence search [11], [12], [35].

For the sake of simplicity and accuracy, many algorithms assume that images are taken by a rectified set of cameras [13], [14]. Consecutively, corresponding fragments of a given image can be found on the same horizontal line in the remaining images.

Some algorithms use three views (left, central and right) [15], [17], [16], [36] as inputs and produce disparity map or depth map for the central view (Fig. 1). Often when it is not important which of left or right view is referred to, a name "side view" is used instead.

During disparity estimation, for a given fragment of the central view, the algorithm searches for the corresponding fragment in the side views that represent the same fragment/portion of the scene.

The correspondence search is done on the basis of Similarity Metric which expresses how probable it is that a certain fragment of one image is the corresponding fragment of the second image. Although the metric used is often called similarity,

it actually expresses dissimilarity between fragments. There are many Similarity Metrics known from literature: Sum of Absolute Difference (SAD) Sum of Squared Difference (SSD), Rank, Census, Cross Correlation and other [3], [4].

The correspondence search is often defined as an optimization problem in which for every fragment of the central image the best (the most similar) fragment of the side views is selected. This optimization problem maybe expressed in terms of energy function using Markov Random Field (MRF) and optimized via one of the optimization algorithms such as Belief Propagation [19], [20], Dynamic Programming [22], or Graph Cut [21].

Since input videos are captured by multiple cameras with different positions, some parts of the observed scene can be occluded, and thus not visible, within some of the views. Disparity estimation for those fragments of a scene is challenging and requires special care. If the the algorithm do not properly taking into account, possible occlusions within the scene, estimated disparity can be wrong. Estimated disparity can indicated not truly corresponding fragments.

In this paper a novel approach to occlusion handling designed to work in three-view disparity estimation algorithms is proposed.

II. OCCLUSION PROBLEM IN DISPARITY ESTIMATION

Given three images, center I_C , left I_L and right I_R of the same size, we search for such a displacement t for every pixel P of center view (at coordinates (x, y)) that minimize cost function expressing a similarity between pixel P (or small fragment around the pixel P like block) and a correspondent pixel P' (small fragment around pixel P') displaced by t in side views (at positions $(x+t, y)$ in left and $(x-t, y)$ in right view). Such displacement is then a disparity of a given pixel P of a center view.

$$d_{Center}(x, y) = \min_t Cost(x, y, t); \quad (2)$$

In disparity estimation based on I from three views (see Fig. 2), a given point of the scene visible from center view can be visible from both of the side views (point A), or only from one of the side views (left or right, point B), or from neither of them (point C).

If the given fragment of the scene visible from center view is not visible from one or both of the side views, we say that

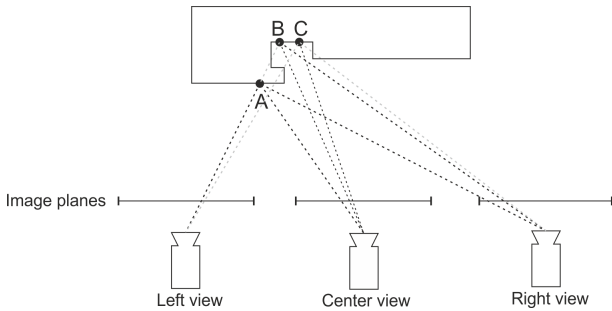


Fig. 2. Occlusion in three-view disparity estimation problem.

a given fragment of the scene is occluded in side view (is not visible from that particular side view).

The simplest method for detecting occluded fragments is cross-checking [23]. Cross-checking tests the consistency of estimated disparity value for pixels from center view with those estimated for pixels in left and right views. If the disparity value estimated in each view is different for a correspondent triple of pixels from center, left and right views given pixels are assumed to be occluded. Next, the disparity value for occluded pixels are extrapolated from neighboring pixels that are not occluded. In order to perform cross-checking, disparity maps for all of the three views are required. Estimation of three disparity maps is not always possible. Even if the estimation of three disparity maps instead of one is possible it is resource and time consuming.

Occlusion handling is performed by adding/putting additional constraints, such as ordering constraint or uniqueness constraint to objective function of optimization procedures like Graph Cut (GC), Dynamic Programming (DP) or Belief Propagation (BP) used to estimate the disparity map.

The ordering constraint [24] imposes the same order of corresponding pixels in all views. If a pixel A is on the left of pixel B in the center view, in the side view pixel A' that is a corresponding pixel of pixel A must be as well on the left of pixel B', a corresponding pixel of pixel B. In real scenes the ordering constraint can be violated in the case of big perspective change or in case of thin objects. In such cases ordering constraint can introduce errors in estimated disparity maps.

The uniqueness constraint [25], [26] imposes the one-to-one correspondence between pixel in center and side views. If a given pixel A of the central view is assigned to a corresponding pixel B in the side view, no other pixel of central view can be assigned to a correspondence with pixel B in side view. This way unique pixel to pixel correspondence is forced across all of the views.

There are many disparity estimation algorithms known, that handle occlusion in efficient way [5], [28], [26]. The main drawback of all of those algorithms are additional constraints (terms) imposed in optimization procedures with increased complexity and thus execution time of the disparity estimation.

Another approach to occlusion handling is to change cost term (eq. 2) composed of similarity metric in optimization algorithms. As we search for corresponding fragment of a central view in both side views simultaneously, there are many ways of defining a $Cost(x, y, t)$ function.

Commonly [17], [16] it is the sum of similarity metrics between a fragment in the center view and corresponding fragments in left and right views.

$$Cost(x, y, t) = Similarity(I_C(x, y), I_L(x+t, y)) + Similarity(I_C(x, y), I_R(x-t, y)) \quad (3)$$

Because of the occlusions Tanimoto [15] proposed to pick just the most similar fragment from either left or right view. The intuition is that the occluded fragment of the images will lead to less similar fragment, thus the minimum of similarity metrics from left and right view is used.

$$Cost(x, y, t) = \frac{NotOcc_L(x, y, t) \cdot Sim(I_C(x, y), I_L(x + t, y)) + NotOcc_R(x, y, t) \cdot Sim(I_C(x, y), I_R(x - t, y))}{NotOcc_L(x, y, t) + NotOcc_R(x, y, t)} \quad (5)$$

$$Cost(x, y, t) = \min(\begin{matrix} Similarity(I_C(x, y), I_L(x + t, y)), \\ Similarity(I_C(x, y), I_R(x - t, y)) \end{matrix}) \quad (4)$$

In this paper we propose yet another way to define cost function which takes into account an occlusion possible within the scene.

III. PROPOSED OCCLUSION HANDLING

As it was said before, a given fragment of a scene visible from center view can be occluded in one or both side views (left or/and right) (Fig. 2). In such a case, searching for a correspondence of a given pixel of center view in this particular side view (left or right) is pointless, as the given fragment of the scene is not visible from that particular side view. Considering the correspondence with an occluded fragment of an image could cause errors in estimated disparity.

Therefore the correspondence search should be performed only in side views in which a considered fragment of a center view is not occluded. The cost function should be constructed in such a way that it considers only similarity metrics from not occluded views. If a given fragment is visible in both views, then the cost function should be an average of both similarity metrics, in order to reduce the influence of noise, which is present in all views. We propose to define the cost function in a way that it considers only similarity metrics of fragments from a not occluded view (either left or right) (eq. 5) where $NotOcc_L(x, y, t)$, $NotOcc_R(x, y, t)$ expresses whether a given pixel of a center view is not occluded in left and right views respectively. Depending on the existence of occlusion in the views, the sum $NotOcc_L(x, y, t) + NotOcc_R(x, y, t)$ in the denominator of eq. 5 can be 2 if a pixel is not occluded in both views, 1 if it is occluded in one of the side views (either left or right), and 0 if it is occluded in both side views. If a given pixel is occluded in both side views the equation 5 loses

its meaning, thus in such a case constant penalty value is used as a cost value.

$$Cost(x, y, t) = const \quad (6)$$

But why a given fragment (object A) of a scene is not visible in a side view? Because in a side view that fragment is occluded by some other part of the scene (object B). Object B blocks light rays from object A, so in side view closer object B is visible instead the farther object A.

Consider the example on Fig. 3 where two points A and B are observed by two cameras (left and center). Point B is closer to the cameras and point A is farther. Point B is visible in both views (left and center) at pixel position B_{Left} and B_{Center} respectively. But due to the occlusion, point A is visible only in center view at pixel position A_{Center} . If there would be no point B, point A should/would be visible in left view at pixel position A_{Left} . The disparity of point B in left view is the difference of the pixel position B_{Left} and B_{Center} and disparity of point A in left view would be (if the point was/would be visible) the difference of pixel position A_{Left} and A_{Center} .

$$d_{Left}(B_{Left}) = B_{Left} - B_{Center} \quad (7)$$

$$d_{Left}(A_{Left}) = A_{Left} - A_{Center} \quad (8)$$

The distance to the camera z is reciprocal to disparity. So a fragment of an image representing a closer object (point B) has bigger disparity than the fragment representing farther object (point A).

$$z_{Left}(A_{Left}) > z_{Left}(B_{Left}) \Leftrightarrow d_{Left}(A_{Left}) < d_{Left}(B_{Left}) \quad (9)$$

For a given pixel A_{Center} of center view at coordinates (x, y) and considered displacement t , corresponding pixel A_{Left} in left view should be at coordinates $(x + t, y)$. So, if we want to check whether a fragment A of a scene is occluded in left view we have to check the disparity (distance) assigned to the considered corresponding pixel A_{Left} in left view. If a disparity $d_{Left}(x + t, y)$ assigned already to considered corresponding pixel A_{Left} is bigger than the considered displacement t then probably a pixel A_{Left} is not a fragment of the same object A but rather some other closer object B that occludes object A in the left view.

Based on such a consideration we can create a function assessing whether for a pixel at coordinates (x, y) and displacement t , corresponding pixel is/can be/will be occluded or not in left and right views.

$$NotOcc_L(x, y, t) = \begin{cases} 1 & \text{for } t \geq d_{Left}(x + t, y) \\ 0 & \text{for } t < d_{Left}(x + t, y) \end{cases} \quad (10)$$

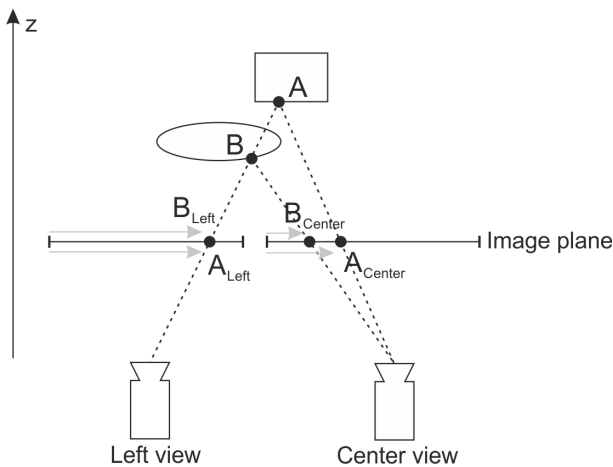


Fig. 3. Occlusion problem in correspondence search.

TABLE I
POSITIONS OF VIEWS USED FOR EVALUATION OF QUALITY OF ESTIMATED DISPARITY MAPS.

Sequence Name	View A	View B	View V
Poznan Street	3	5	4
Poznan Hall 2	5	7	6
Poznan CarPark	3	5	4
Book Arrival	7	9	8

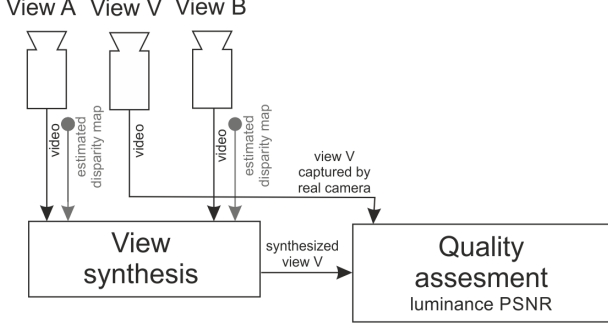


Fig. 4. Disparity map quality evaluation methodology.

$$NotOcc_R(x, y, t) = \begin{cases} 1 & \text{for } t \geq d_{Right}(x - t, y) \\ 0 & \text{for } t < d_{Right}(x - t, y) \end{cases} \quad (11)$$

$NotOcc(x, y, t)$ equal 1 means that the corresponding pixel in side view at a given displacement is probably not occluded.

IV. APPLICATION OF PROPOSED IDEA

The proposed idea is general - it does not impose any particular source of disparity maps d_{Left} for left and d_{Right} for right view. But in general, disparity maps for left and right views are unknown before estimating the disparity for central view.

Commonly, disparity maps are estimated iteratively with the use of such algorithms like Belief Propagation or Graph Cut. In such algorithms, at each iteration of the estimation, algorithm maintains up-to-date / best already estimated disparity map for center view. This disparity map is further refined in the next iteration of the algorithm.

For our occlusion detection we propose to use disparity maps of side views created based on the disparity map of center view through Depth-Image-Based Rendering (DIBR). After each iteration of a disparity estimation algorithm, we create disparity maps of side views (d_{Left} and d_{Right}) from the best already estimated disparity map of a center view. This way if the estimation algorithm used assigned already some disparity $d_{Center}(B)$ to some pixel B_{Center} , then pixel A_{Center} cannot have such a disparity that the corresponding pixel A_{Left} (Fig. 3) is at the same position as corresponding pixel B_{Left} of pixel B_{Center} . In other words fragment B of a scene represented by pixel B_{Center} in center view should occlude a fragment A of a scene (represented by pixel A_{Center} in center view) seen from left view.

V. EXPERIMENTS

We have implemented our idea in Depth Estimation Reference Software (DERS) [18] version 5.0 developed by Moving

Picture Experts Group (MPEG) of International Standardization Organization (ISO) during works on 3D video compression standardization. DERS is the state-of-the-art disparity estimation technique, designed with 3D video application in mind. It uses Graph Cut as the optimization algorithm along with many other techniques that improve or/and speed up disparity estimation from three input videos.

Proposed approach was tested on four 3D video test sequences recommended by the MPEG committee (Fig. 5) namely: Poznan Street, Poznan CarPark, Poznan Hall 2 [1], Book Arrival [2].



(a) Poznan Street [1]

(b) Poznan CarPark [1]



(c) Poznan Hall2 [1]



(d) Book Arrival [2]

Fig. 5. Exemplary frames from multiview test sequences used in experiments

In applications such as Free View Television, disparity maps are used mainly for the purpose of view synthesis. Therefore, we have evaluated our proposed method indirectly, by assessing the quality of the synthesized views.



(a) Tsukuba

(b) Venus



(c) Teddy



(d) Cones

Fig. 6. Standard Middlebury dataset [29] used for evaluation of proposed algorithm

Disparity maps for two views A and B (Fig. 4) have been estimated with the use of the proposed method and original unmodified DERS software. Based on views A and B and estimated disparity maps for views A and B, view V that is positioned in between of view A and B was synthesized. Exact view numbers for each of test sequences used during experiments are provided in Table I.

The quality of estimated disparity maps for views A and B is measured as a quality of rendered view V. Quality of synthesized view V is expressed by PSNR of luminance in comparison with view V captured by real camera positioned at the same spatial position (see Fig. 4).

Such methodology is compliant with experimental methodology developed and approved by the MPEG committee of International Standardization Organization and is used by other research institutes, targeted at high quality 3D television for e.g. autostereoscopic displays.

In the course of evaluation disparity maps were estimated for every frame within the sequences (mostly 250 frames per view). This has allowed to evaluate our algorithm on a wide range of different images. The disparity estimation was done with pixel, half-pixel and quarter-pixel precision. Also, a wide range of regularization terms used in Graph Cut algorithm has been evaluated. In DERS the regularization is controlled by so-called smoothing coefficient. In experiments, the range of 1 to 4 was explored.

We have also evaluated our algorithm on standard Middlebury dataset [29]: Tsukuba, Venus, Teddy and Cones (Fig. 6). In the course of that, we have modified DERS algorithm to directly output raw disparity maps in the format required by Middlebury evaluation webpage [30]. Because both proposed methods and the DERS algorithm are designed to work with three input images, we have extended recommended/standard stereo pair with third image as specified in Table II.

TABLE II
SPECIFICATION OF THREE VIEWS USED FOR DISPARITY ESTIMATION FOR EACH MIDDLEBURY DATASET.

Dataset name	Additional view	Standard stereo pair	
	Left view	Center view	Right view
Tsukuba	2	3	4
Venus	0	2	6
Teddy	0	2	6
Cones	0	2	6

TABLE III
QUALITY COMPARISON BY PSNR OF A SYNTHESIZED VIEW FOR THE BEST DEPTH MAPS WITH RESPECT TO SMOOTHING COEFFICIENT.

Sequence Name	Pixel precision			Half-pixel precision			Quarter-pixel precision		
	DERS [dB]	Proposed [dB]	Gain	DERS [dB]	Proposed [dB]	Gain	DERS [dB]	Proposed [dB]	Gain
Poznan Street	36.31	37.41	1.10	36.78	37.70	0.92	36.96	37.88	0.90
Poznan Hall2	34.62	36.06	1.44	34.62	36.11	1.48	34.74	36.39	1.56
Poznan CarPark	31.71	33.89	2.18	31.36	33.87	2.50	31.51	33.99	2.48
Book Arrival	36.06	36.36	0.30	37.37	37.38	0.02	37.37	37.39	0.02
Average	-	-	1.26	-	-	1.23	-	-	1.24

VI. RESULTS

The comparison of quality of estimated disparity maps for proposed method versus original DERS can be found in Fig. 7d, 7a, 7c, 7b. As it can be noticed, the smoothing coefficient can have significant impact on the quality of disparity maps estimated by DERS. It can be expected that in a real-world-use scenario, this parameter will be automatically controlled to provide the best results. Therefore, in summarized Table III, we have presented only the best-performing cases. Depending on the case, the proposed occlusion handling brings a gain of 0.02-2.50 dB of luminance PSNR of synthesized view, related to the original unmodified DERS. On average, the proposal provides an improvement of 1.26 dB for pixel-precise disparity estimation, 1.23 dB for half-precise disparity estimation, and 1.18 db for quarter-precise disparity estimation.

The application of proposed occlusion handling to Middlebury images results in 0.2 bad pixel improvement (Table IV). Please keep in mind that Middlebury datasets have very little occlusions.

VII. CONCLUSION

We have presented a novel approach to occlusion handling in disparity estimation, based on a modification of similarity cost function. Proposed approach has been tested in the three-view disparity estimation scenario. For occlusion detection synthesized disparity maps of left and right views have been used.

For well-known multiview video test sequences, the experimental results show that the proposed approach provides virtual view quality improvement of 1.25 dB of luminance PSNR over the state-of-the-art technique implemented in MPEG Depth Estimation Reference Software (DERS). Moreover, direct quality evaluation of estimated disparity reveals that proposed the approach reduces a number of bad pixels by 1.26 p.p.

ACKNOWLEDGMENT

Research project was supported by National Science Centre, Poland, according to the decision DEC-2012/05/N/ST7/1279.

REFERENCES

- [1] M. Domański, O. Stankiewicz, K. Wegner et al., *Pozna multiview video test sequences and camera parameters*, ISO/IEC JTC1/SC29/WG11 Doc. M17050, Xian, China, October 2009.

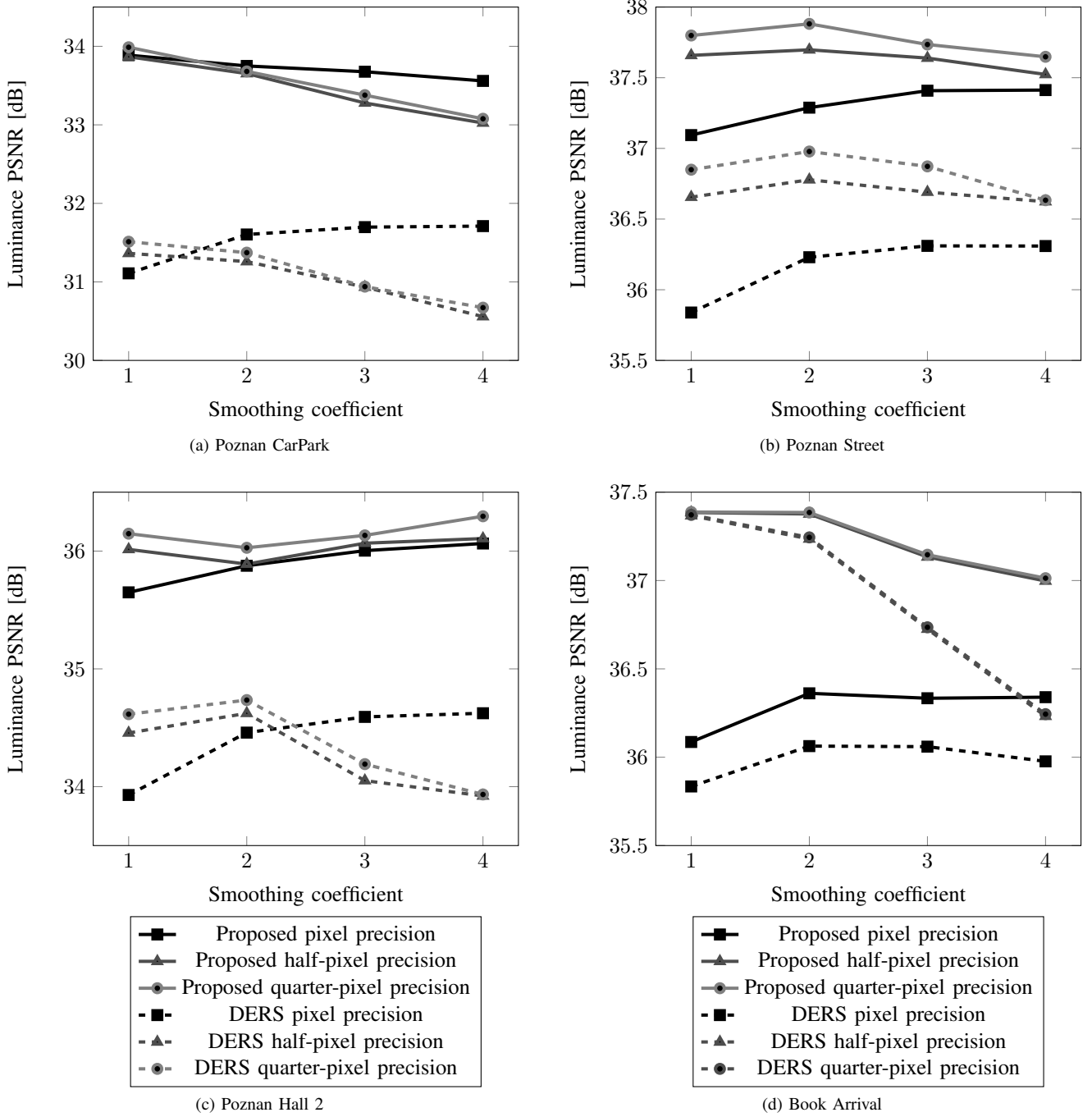


Fig. 7. Performance comparison of depth estimation with use of proposed method and DERS

TABLE IV
COMPARISON OF PROPOSED METHOD WITH DERS ON MIDDLEBURY DATASETS.

Algorithm	Tsukuba			Venus			Teddy			Cones		
	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc
GC+occ	1.19	2.01	6.24	1.64	2.19	6.75	11.2	17.4	19.8	5.36	12.4	13.0
2OP+occ [36]	2.91	3.56	7.33	0.24	0.49	2.76	10.9	15.4	20.6	5.42	10.8	12.5
Putv3	1.77	3.86	9.42	0.42	0.95	5.72	7.02	14.2	18.3	2.40	9.11	6.56
CostAggr+occ [37]	1.38	1.96	7.14	0.44	1.13	4.87	6.80	11.9	17.3	3.60	8.57	9.36
DERS	2.70	3.30	12.10	0.67	1.25	8.53	10.2	11.5	23.3	5.17	7.33	9.50
Proposed	2.65	3.01	11.20	0.63	1.02	8.34	9.96	10.97	—	5.02	7.12	—

- [2] I. Feldmann, A. Smolic, et al., *HHI Test Material for 3D Video*, ISO/IEC JTC1/SC29/WG11, Doc. M15413, Archamps, France, 2008.
- [3] K. Wegner, O. Stankiewicz, *Similarity measures for depth estimation*, 3DTV-Conference 2009, Potsdam, Germany, May 2009.
- [4] S. Birchfield and C. Tomasi, *A pixel dissimilarity measure that is insensitive to image sampling*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(4):401406, April 1998.
- [5] Woo-Seok Jang, Yo-Sung Ho, *Efficient disparity map estimation using occlusion handling for various 3D multimedia applications*, IEEE Consumer Electronics, vol. 57, no. 4, pp.1937,1943, November 2011.
- [6] K. Miller, P. Merkle, T. Wiegand, *3-D video representation using depth maps*, Proceedings of the IEEE, vol. 99, no. 4, April 2011.
- [7] L. Zhang and W. J. Tam, *Stereoscopic image generation based on depth images for 3DTV*, IEEE Trans. on Broadcasting, vol. 51, no. 2, pp. 191-199, June 2005.
- [8] S. Y. Kim, J. H. Cho, and A. Oschan, *3D video generation and service based on a TOF depth sensor in MPEG-4 multimedia framework*, IEEE Trans. on Consumer Electronics, vol. 56, no. 3, pp. 1730-1738, August 2010.
- [9] M. Domański, A. Dziembowski, A. Kuehn, M. Kurc, A. Łuczak, D. Mieloch, J. Siast, O. Stankiewicz and K. Wegner, *Experiments on acquisition and processing of video for free-viewpoint television*, 3DTV Conference 2014, Budapest, Hungary, July 2014.
- [10] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed., Cambridge University Press, 2003, pp. 262-278.
- [11] M. Okutomi and T. Kanade, *A multiple baseline stereo*, IEEE Trans. PAMI, 15(4):353363, April 1993.
- [12] R. T. Collins, *A space-sweep approach to true multi-image matching*, In CVPR96, pp. 358363, San Francisco, 1996.
- [13] J. Stankowski, K. Klimaszewski, O. Stankiewicz, K. Wegner, M. Domański, *Preprocessing methods used for Poznan 3D/FTV test sequences*, ISO/IEC JTC1/SC29/WG11 MPEG 2010/M17174, Doc. m17174, Kyoto, Japan, January 2010.
- [14] J. Stankowski, K. Klimaszewski, Book: "Image Processing and Communications Challenges 2", Chapter: "Application of epipolar rectification algorithm in 3D Television", Advances in Intelligent and Soft Computing: Vol. 84, Springer-Verlag, Berlin, 2010, pp. 345-352, ISBN: 978-3-642-16294-7.
- [15] M. Tanimoto, T. Fujii, K. Suzuki, N. Fukushima, and Y. Mori, *Reference softwares for depth estimation and view synthesis*, ISO/IEC JTC1/SC29/WG11, Doc. M15377, Archamps, France, April 2008.
- [16] Sang-Beom Lee, Yo-Sung Ho, *Multi-view Depth Map Estimation Enhancing Temporal Consistency*, 23rd International Technical Conference on Circuits/Systems, Computers and Communications.
- [17] M. Wildeboer, N. Fukushima, T. Yendo, M. T. Panahpour, T. Fujii, M. Tanimoto, *A semi-automatic multi-view depth estimation method*, Proceedings of the SPIE, Volume 7744, 2010.
- [18] M. Wildeboer, O. Stankiewicz, K. Wegner, *A soft-segmentation matching in Depth Estimation Reference Software (DERS) 5.0*, ISO/IEC JTC1/SC29/WG11 Doc. M17049, Xian, China, October 2009.
- [19] J. Sun, N. N. Zheng, and H. Y. Shum, *Stereo matching using belief propagation*, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 25, no. 7, pp. 787-800, July 2003.
- [20] P. Felzenszwalb and D. Huttenlocher, *Efficient belief propagation for early vision*, In Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp.261-268, June 2004.
- [21] Y. Boykov, O. Veksler, and R. Zabih, *Fast approximate energy minimization via graph cuts*, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 23, no. 11, pp.1222-1239, November 2001.
- [22] O. Veksler, *Stereo correspondence by dynamic programming on a tree* CVPR 2005.
- [23] G. Egnal and R. Wildes, *Detecting binocular halfocclusions: empirical comparisons of five approaches*, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 24 no. 8, pp. 1127-1133, August 2002.
- [24] A. Bobick and S. Intille, *Large occlusion stereo*, Int. Journal of Computer Vision, vol.33, no. 3, pp. 181-200, September 1999.
- [25] D. Marr and T. A. Poggio, *Cooperative computation of stereo disparity*, Science, vol. 194, no. 4262, pp. 283-287, October 1976.
- [26] V. Kolmogorov and R. Zabih, *Computing visual correspondence with occlusions using graph cuts*, In Proc. IEEE International Conference on Computer Vision, pp. 508-515, July 2001.
- [27] T. Liu, P. Zhang, and L. Luo, *Dense stereo correspondence with contrast context histogram, segmentation-based two-pass aggregation and occlusion handling*, Lecture Notes in Computer Science, vol. 5414, pp. 449-461, January 2009.
- [28] R. Ben-Ari and N. Sochen, *Stereo matching with Mumford-shah regularization and occlusion handling*, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 32, no. 11, pp. 2071-2084, November 2010.
- [29] D. Scharstein and R. Szeliski, *High-Accuracy Stereo Depth Maps Using Structured Light*, In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003), volume 1, pages 195-202, Madison, WI, June 2003.
- [30] Middlebury Stereo Evaluation - Version 2, webpage visited 2015-01-24, <http://vision.middlebury.edu/stereo/eval>.
- [31] G. Tech, K. Wegner, Y. Chen, S. Yea, *3D-HEVC Draft Text 6* Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 Doc. JCT3V-J1001, 10th Meeting: Strasbourg, FR, 1824 October 2014.
- [32] Annex I *Multiview and Depth video coding* of ISO/IEC 14496-10, Int. Standard *Generic coding of audio-visual objects Part 10: Advanced Video Coding*, 8th Ed., 2013, also: ITU-T Rec. H.264, Edition 8.0, 2013.
- [33] *3D-AVC Draft Text 9*, JCT-3V of ITU T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, Doc. JCT3V-G1003, San Jose, USA, 2014.
- [34] D. Scharstein and R. Szeliski, *A taxonomy and evaluation of dense two-frame stereo correspondence algorithms*, International Journal of Computer Vision, 47(1/2/3):7-42, April-June 2002.
- [35] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, *A comparison and evaluation of multi-view stereo reconstruction algorithms*, in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR06), pp. 519526, June 2006.
- [36] O. Stankiewicz, "Stereoscopic depth map estimation and coding techniques for multiview video systems", PhD Dissertation at Poznan University of Technology, Faculty of Electronics and Telecommunications, 2014.